X3S37-77-56

Source: U.S. Department of Defense
Advanced Research Projects Agency
and Defense Communications Agency

Title: Datagram Interface Requirements for Packet Networks

Date: July 22, 1977

Abstract: This paper discusses the motivation for augmenting the X.25 packet network interface standard to include a datagram mode of operation. Technical arguments based on forseeable commercial and military requirements are offered.

DATAGRAM INTERFACE REQUIREMENTS FOR PACKET NETWORKS

Background

Packet communication networks are either in operation or under construction in at least 20 countries, and, in some countries, more than one network is to be found. Commercial interest in these networks is increasing and both public and private networks have been developed. A natural consequence of the development of these networks is the desire to establish standards for interfacing to them. Such standards would permit computer equipment manufacturers to build and support compatible software and hardware and could also, potentially, ease the ultimate task of interconnecting many of these networks to each other.

Over the past 4 years, packet network interface standards have been the subject of many lively debates, both in the international commercial and government telecommunications world and also in academic circles.

This paper discusses the motivation for augmenting the X.25 packet network interface standard to include a datagram mode of operation. The following conclusions are drawn:

1. X.25 cannot concurrently support transaction services involving arbitrarily large numbers of sources and destinations.

2. X.25 places terminal handling functions at the wrong protocol layer.

3. Dual-homing of hosts and gateways is make-shift at best under the virtual circuit strategy.

4. The X.25 network interconnection strategy does not accommodate non-sequencing or lossy nets.

5. Flexible network interconnection with dynamic alternate gateway routing is unsupported by X.25.

6. End-to-end encryption services may fail if the X.25 interface arbitrarily recombines packets before delivery to a host. Packet fragmentation and reassembly under X.25 appears to lose important end-end boundary information.

7. Packet broadcasting is inefficient at best and impossible, at worst, if virtual circuits are used to achieve it.

8. Real-time applications requiring low delay but not requiring guaranteed or sequenced data delivery will be inefficient and possibly inoperable through X.25 virtual circuits.

9. Regardless of the use of virtual circuits or datagrams, host level protocols must provide sequencing, retransmission, duplicate detection, fragmentation, reassembly, and flow control techniques to assure reliable and controlled host-level intranet and internet communication.

Many of these conclusions stem from military network requirements which are more general than those for public nets. A counter-intuitive conclusion is that the military requirements impose fewer restrictions on network operation than the X.25 interface does, admitting a wider class of nets to the internetting community.

Network Models

Most views of packet networks distinguish between "datagram networks" and "virtual circuit" networks. In fact, this is a very misleading taxonomy. A more accurate picture can be obtained by separating the network interface characteristics from internal network operation.

For example, the interface to the ARPANET is datagram-like. No "connection set-up" exchanges occur between subscriber and network to cause packets to flow from source to destination. The network has two modes of internal operation, however. Packets labeled "type 0" are delivered in sequence to the destination. The mechanism to do this is hidden within the network and does not require knowledge of subscriber-level connections. Type 0 packets flowing between pairs of source and destination hosts are kept in sequence. The second mode of operation is pure datagram. "Type 3" packets are not sequenced and delivery is not guaranteed. The user selects these services by packet labelling and not by connection set-up.

This combination of interface and internal operation may be contrasted with the U.S. TELENET operation which requires an external connection set up request and acknowledgement before a subscriber can send data to a remote site.* In this mode of operation, the set up packet traverses the network and a record of the connection is made at the destination (but not at intermediate points). A return packet confirms the virtual circuit set-up. Subscribers typically assign a short identifier to the virtual circuit to avoid sending the entire source and destination address in each packet from DTE to DCE (i.e., subscriber to network).

The internal operation of the Telenet subnet is otherwise similar to that of ARPANET. Packets carry both source and destination addresses and are routed through the store and forward network in an adaptive fashion. Re-sequencing occurs at the destination DCE, as in ARPANET.

A third example may be found in the TYMNET where subscribers are assigned explicit fixed routes through the network. The connection set-up is managed centrally by a supervisory computer which sends specific routing information to each switching node in the path from source to destination. Each character of text (or group of characters) is associated with a source "line number" which identifies the source subscriber. A "Frame" of text is made up of groups of characters from one or more terminals. The groups are each separately routed from node to node, so the frame, when it arrives, is check-summed and disassembled. New frames are constructed at each node based on routing tables. By virtue of the fixed routing and link control procedures, all data remains in sequence from source to destination.

*This does not rule out efficiencies such as allowing the connection set-up packet to carry data as well.

Finally, we can consider the CYCLADES network built at the French Institute Recherche d'Informatique et d'Automatique (IRIA). In this net, no attempt is made to maintain the order of packet arrivals at the destination. No "circuit set-up" packets are needed. Instead, as in ARPANET, each subscriber packet contains a source and destination address which can be used to route the packet through the network. Autodin II has similar characteristics.

A reasonable taxonomy for these nets is:

1. Datagram Interface, Virtual Circuit Operation (e.g., ARPANET type 0 packets)

2. Virtual Circuit interface, Virtual Circuit operation (e.g., Telenet, EPSS, TRANSPAC, DATAPAC, TYMNET)

3. Datagram Interface, Datagram operation (e.g., CYCLADES, AUTODIN II, ARPANET type 3 packets)

The obvious fourth combination (Virtual circuit interface, datagram operation) is possible but is not, to the author's knowledge, in use anywhere. Such a net would require that subscribers set up the ends of a virtual circuit at source and destination DCE's, but would not attempt to deliver packets in sequence.

The builders of public or commercial packet networks have been and continue to be motivated by the marketplace. The typical customer of a packet network is looking for lower cost alternatives to leased lines and is not very interested in building special software simply to accommodate packet network operation. A predictable (though not necessarily desirable) consequence of this replacement strategy is that the replacement technology offer equivalent or nearly equivalent service at lower cost. Other arguments favoring the virtual circuit interface/operation mode appear when typical existing computer network applications are examined. Most of these applications involve time-sharing, remote job entry and bulk file transfer services requiring the equivalent of dial-up, hard-wire, or leased lines. It is not surprising to find that most commercial or public packet switching services have been tailored to be attractive to existing applications.

Virtual circuits are not real circuits, of course. For one thing, they exhibit variable delay and for another, they usually require that the subscriber "packetize" his traffic. Handling terminal traffic is an exception since most existing terminals are serial, character-oriented devices. Thus, packet assembly and disassembly (PAD) functions are often offered by packet network service suppliers to support terminal access to time-sharing systems. Speed and character conversion services can easily be incorporated into the PAD devices as well, making the packet service even more attractive.

Many issues in packet network interfacing can only be understood in the context of computer communication protocol design. In the next section, a model of protocol architecture is offered as an aid to thinking about interfaces. The model is not new, having been developed at least by 1968, when the ARPANET was being constructed, and most likely, long before that.

Protocol Layering

Figures 1 and 2 illustrate the basic model of packet network structure upon which our arguments are based. In figure 1, we show a network structure which places PAD functions outside the packet network boundary and above the host level of protocol. Of course, this does not mean that suppliers of packet service cannot or should not offer PAD services, but simply that PAD services should reside outside the host/network interface layer. We will return to this point when we discuss virtual terminal protocols. Figure 1 also shows "gateways" between networks. We will discuss this in a later section on packet network interconnection.

Figure 2 offers a conceptual view of the layers of protocol and interfaces we can expect to find in packet networks. The layers are connected vertically by real interfaces which carry data and control physically from one layer to the next. The layers are connected horizontally by logical or conceptual paths which constitute virtual interfaces carrying embedded control and data.

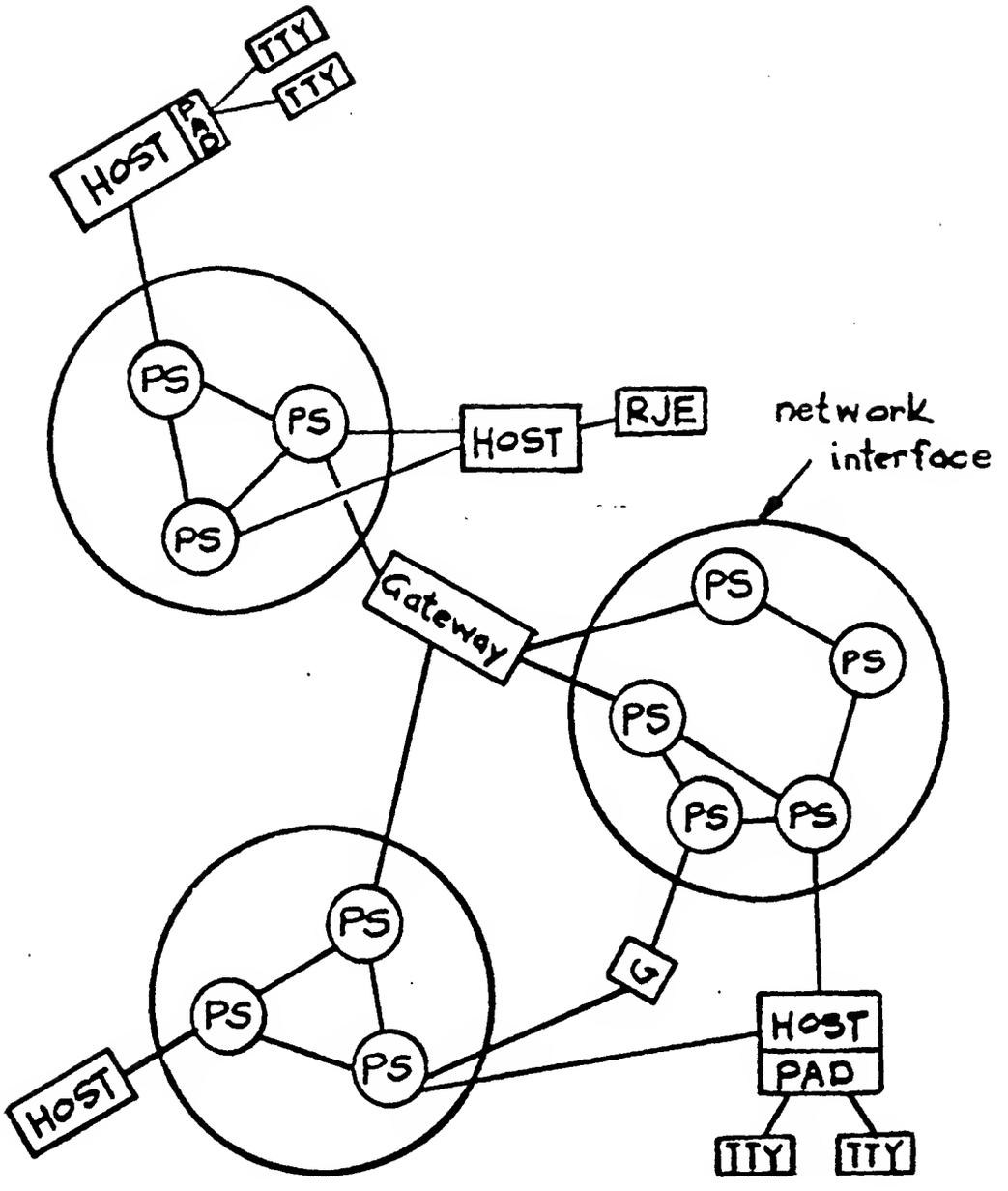
The conceptual embedding is illustrated in figure 3. In real networks, of course, some optimizations can be found which avoid the need to actually carry in each packet all the control needed.

Generally speaking interfaces define ways of carrying (and distinguishing) control information and data. Physical interfaces may use different paths to distinguish control from data (e.g., control and data lines on an X.21 interface). Virtual interfaces usually employ formatting to embed control and data in the same transmission medium.

The CCITT recommendation X.25, as apparently used in TELENET, EPSS, DATAPAC, and TRANSPAC, imposes an unnecessary restriction on subscribers with respect to protocol layering. Virtual circuits are identified with a 12 bit identifier. For transaction applications in which these may be tens or hundreds of thousands of sources or destinations (e.g., point of sale services), the intrusion of virtual circuit numbering on the subscriber seems unnecessary. The minor efficiency obtained through this imposition (shorter DTE to DCE packets) is lost when packets are internally carried with full source and destination addressing anyway, so it is not apparent that much is gained through this design.

Conclusion 1: X.25 cannot concurrently support transaction services involving arbitrarily large numbers of sources and destinations.

The argument that virtual circuits are required to deal with flow control is simply invalid. To be sure, source and destination packet switches must protect their buffer resources. They can do this at the subnetwork level by exercising flow control based on source and destination packet switch pairs or, if desired, source and destination subscriber (i.e., host interface) pairs. There is no reason for the subnet to care about the number of logical connections between internal processes of two subscribers. The ARPANET implementation has even demonstrated that flow control information between source and destination packet switches can be dynamically destroyed and recreated, if need be, to prevent dormant pairs of subscribers from consuming valuable resources in their respective packet switches.



Model of Packet Network Structure

Figure 1

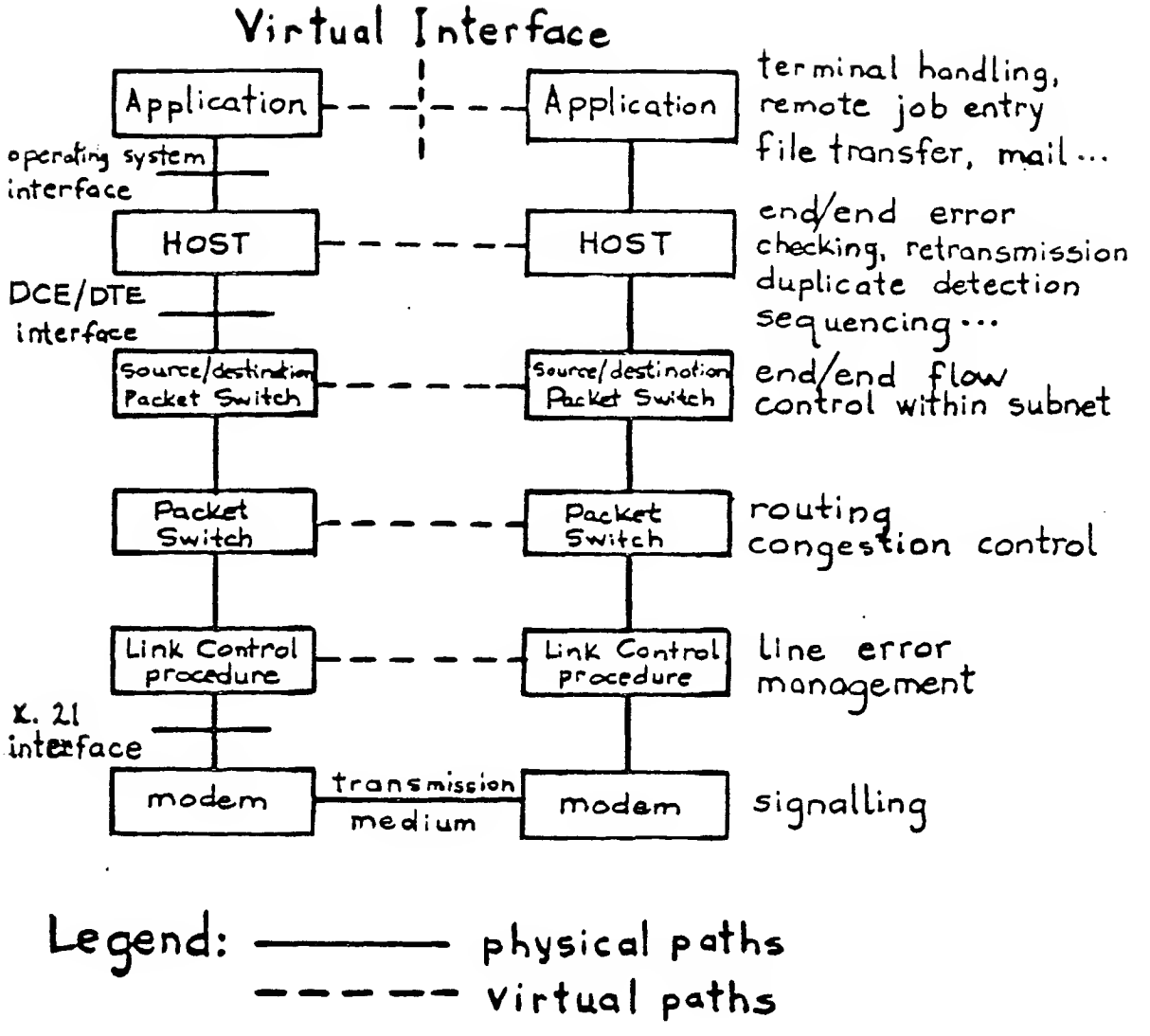
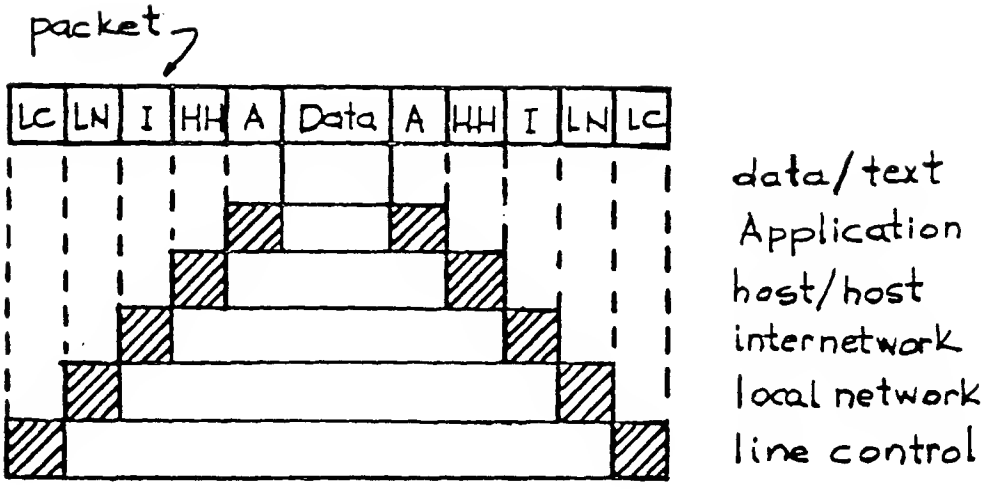

Virtual Interface
Application
Application
terminal handling,
remote job entry
file transfer, mail...
operating system
interface
HOST
HOST
end/end error
checking, retransmission
duplicate detection
sequencing...
DCE/DTE
interface
Source/destination
Packet Switch
Source/destination
Packet Switch
end/end flow
control within subnet
Packet
Switch
Packet
Switch
routing
congestion control
Link Control
procedure
Link Control
procedure
line error
management
X. 21
interface
modem
transmission
medium
modem
signalling
Legend:
physical paths
virtual paths

Figure 2



Embedding of Protocol Control in a typical packet

Figure 3

Virtual Terminals

One of the most important notions arising out of packet network research is the idea that all terminal handling should be accomplished through the use of a virtual terminal standard. A conceptual virtual terminal is defined with certain characteristics such as character set, page size, line length, format control (backspace, line feed, tabs, vertical tabs, end-of-line indicator) and editing functions (character deletion, word deletion, line deletion). Any particular terminal is interfaced to the virtual terminal level of protocol (PAD in figure 1, application level in figure 2). The physical terminal characteristics are translated into the standard virtual terminal representation for transmission through the packet network.

Specific parameters about real terminals may be exchanged through the virtual terminal protocol so that hosts serving virtual terminals can maintain a model of service requirements at the user end. Negotiations to decide which end should echo characters typed or to adapt to terminal line or page size can be conducted using the virtual terminal protocol.

The commercial and public network community has recognized the value of this concept and the X.25 network interface standard provides for special "virtual terminal" virtual circuits. X.25 does not, however, exhibit the layering shown in figure 2. Indeed, this is not surprising. To achieve the layering of that figure, a common host level protocol must be postulated which will support application protocols such as virtual terminals. Since few manufacturers were or are prepared to offer a common host protocol, the network purveyors were forced to "emulate" the host-to-host through the use of virtual circuits which would provide sequencing, flow control, duplicate detection, end-to-end retransmission and so on. It is our view that this is a reasonable short-term solution, but that eventually, these functions must be agreed upon, supplied, and supported by computer equipment manufacturers. This need will become more apparent when we discuss network interconnection.

Conclusion 2: X.25 places terminal handling functions at the wrong protocol layer.

Multihomed Hosts

In figure 1, we have shown one host which is connected to two different packet switches in the same network. If virtual circuit concepts are used to support such hosts, then one of two possible scenarios can be predicted. Either one of the connections is viewed as a primary link, with all traffic specifically addressed to it (the others being unused until the primary link fails) or all links are used, but packets associated with a particular virtual circuit are always delivered over the same link. The virtual circuit concept requires that the network sequence packets before delivery to the host. This can only be achieved if all packets for a given virtual circuit are

re-sequenced at a single destination packet switch.

An alternative strategy which permits packets to be arbitrarily delivered on either link requires that sequencing, duplicate detection, and retransmission occur outside of the network in the host or a front-end which is connected to two or more packet switches.

If the latter idea is pursued, then multihomed hosts need have only one name. The source host need not know that there are two (or more) connections between the destination host and the network, leaving the routing decision to the packet network. It should be obvious that this can be accomplished if a pure datagram facility is offered by the network (i.e., no assumption of sequencing, no binding of packet delivery to any particular host/network link). Virtual circuit methods are restricted to less flexible use of the multiple links.

If virtual circuits are used to serve multihomed hosts, and the primary connection breaks, there is no way for the source packet switch to distinguish between the case that a packet was successfully delivered to the host but the acknowledgement was lost by the broken connection and the case that the packet in fact was not delivered. Thus, any attempt at repair by switching to an alternate link will require host level protocols to sort out duplicates. To properly deal with this case, host-host protocols must employ their own level of sequence numbering, end-to-end acknowledgement, retransmission, and duplicate detection.

Conclusion 3: Dual-homing of hosts and gateways is make-shift at best under the virtual circuit strategy.

Conclusion 4: The X.25 network interconnection strategy does not accommodate non-sequencing or lossy nets.

Network Interconnection

Finally, we must come to grips with the problem of network interconnection. We strongly agree with the designers of X.25 that network interface standards will be of great help in solving this problem. However, we feel equally strongly that the correct set of standard network services required to effect interconnection should be minimized to allow for the maximum range of network implementations and to admit very flexible internetwork routing and error recovery strategies.

In figure 1 we illustrate the conceptual notion of interconnecting networks through a "gateway." It should be kept in mind that the following arguments do not depend on a monolithic device to interconnect two networks, although the notion of a gateway box is one possible realization of the concept. Other possibilities include the implementation of the gateway function in two separate "halves" (or "thirds", if three networks are involved) which might ultimately share the hardware of a packet switch.

One of the most important notions of network interconnection to emerge from research in this area is the concept of terminating or "cauterizing" internal network protocols at the gateway. Since this is analogous to the termination of internal protocols at the point where hosts connect to networks, we are led to the idea that a gateway should interface to a network in the same way a host does. This does not preclude a network from distinguishing between hosts and gateways. To give a simple example, a

gateway might be associated with any number of networks (i.e., the internal routing and addressing functions of the network might be aware of internet addresses; packets containing destination addresses outside the local network could be routed to an appropriate gateway). Hosts would generally have only one name (although the network might also know that more than one interface exists to that host in the multi-homed case).

Packet switching with adaptive routing mechanisms allows the network to recover from internal failures by automatic re-routing of packets around broken links or nodes. This is separable from the notion of re-routing around congested areas although networks like the ARPANET have combined these capabilities in one routing mechanism.

The most fundamental issue concerning network interconnection is gateway reliability. Public and private networks can seek to improve matters by building redundant, multiple processor, multiply-interconnected gateways at a given location.

For military networks, and probably even for commercial and private networks, simply building more reliable gateways is not a sufficient solution. Under hostile conditions, gateways may be destroyed or access to them may be denied (e.g., through jamming in radio nets). Consequently, it is essential that network interconnection strategies naturally allow the adaptive routing of packets through multiple gateways, regardless of the end-to-end "circuit" associated with the packet.

It could be argued that the problems faced by the military are unique and need not have any impact on commercial or public nets. In fact, standards for data communication are of central concern to the military both because equipment and systems are supplied to the military by commercial manufacturers and because, in many countries, military data communication facilities are supplied by the Post, Telephone and Telegraph (PTT) organizations which also supply public facilities.

In the U.S., the Department of Defense is largely required to use commercial facilities whenever possible. It is essential that special military facilities and commercial facilities be compatible. The cost of providing for increased flexibility in network interconnection strategies is negligible and in some cases, could lead to less expensive commercial systems as well.

Arguments have been offered to the effect that networks exhibiting X.25 interfaces can easily be interconnected by "gluing together" virtual circuits at a gateway. If this notion is taken literally, then the gateway becomes a critical link in the internet virtual circuit. At the gateway, status information (flow control, local virtual circuit number - which may be different for each net, local source/destination addresses and perhaps internet source/destination addresses, accounting information, etc.) about the virtual circuit must be kept up-to-date. If the gateway fails or becomes isolated, a new virtual circuit must be created. It does not appear likely that this can be done entirely without the awareness of the subscriber, in particular since it involves the potential loss or duplication of packets which were in the gateway when it failed.

An alternative interpretation of the X.25 internet argument suggests that the gateway can transparently pass packets (in X.25 format) from one net to another and that the ultimate packet switches at the source and destination can cooperate to achieve an end-to-end virtual circuit. It is further argued that the gateway can manipulate the end-to-end flow control mechanism so as to achieve intergateway or source packet switch-to-gateway flow control. This model leads to the conclusion that a gateway must maintain status information about every virtual circuit passing through it. For networks supporting substantial internet traffic, the number of virtual circuits could reach on the order of the square of the number of subscribers.

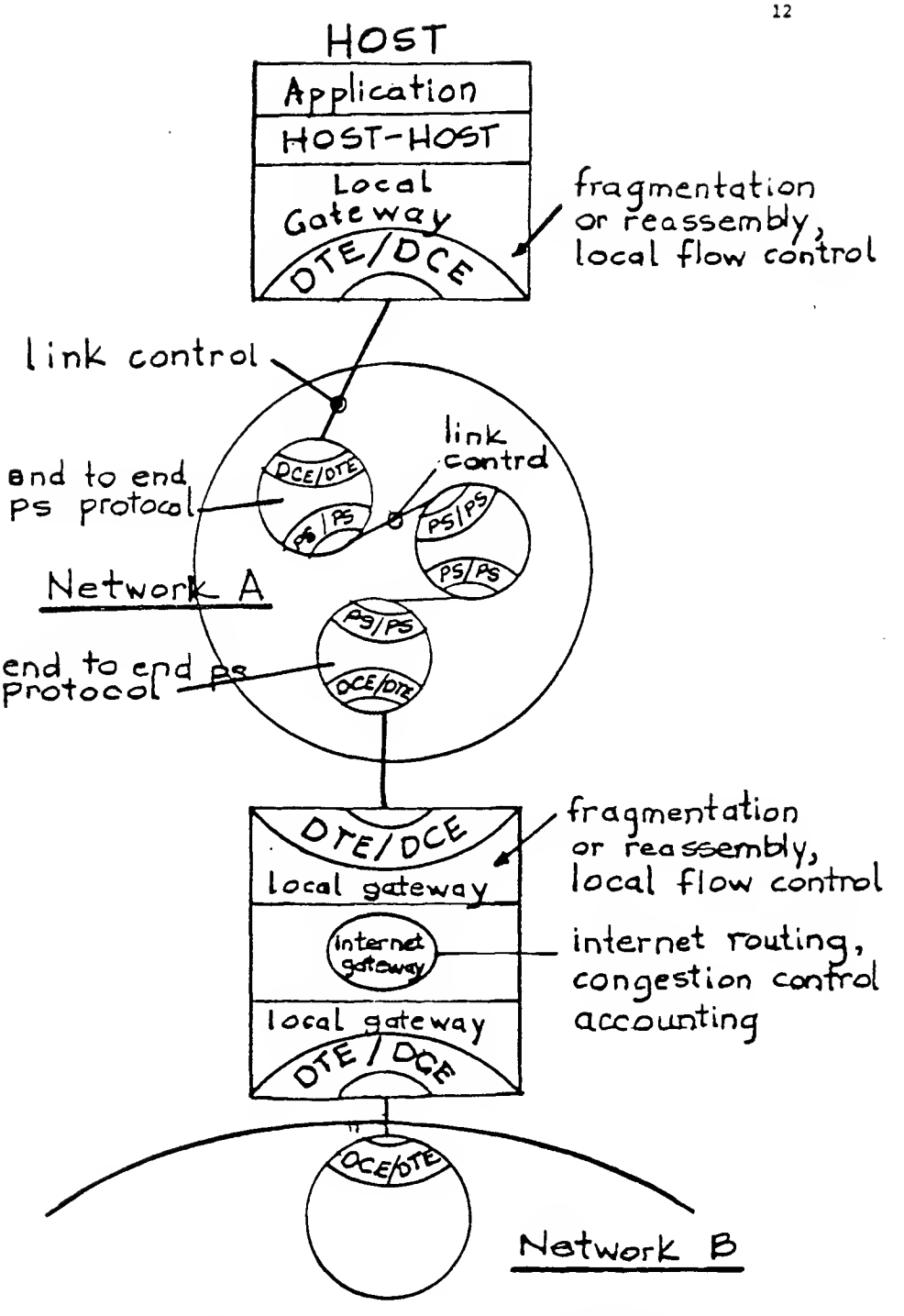
Although international tariff barriers may initially limit the amount of traffic between national networks, it is clear that electronic mail and file transfer types of service will be in high demand. Eventually this will constitute a major share of the internet traffic among all interconnected networks. Any network interconnection strategy which depends on state information in gateways will be subject to major recovery problems after a gateway crash. Other methods of interconnection have been developed and demonstrated which do not require gateways to maintain internal state information about the subscriber-level traffic passing through (except for accounting information which is periodically extracted from the gateway anyway). One example of this interconnection strategy is found in the interconnection of the U.S. ARPANET, Packet Radio Net and Packet Satellite Net through "state-free" gateways.

Conclusion 5: Flexible network interconnection with dynamic alternate gateway routing is unsupported by X.25.

It must also be recognized that not all networks will utilize the same transmission technology. Some will employ satellites, some leased lines of differing rates, some ground radio systems, some involving mobile subscribers, some using twisted pair or coaxial rings, some using multi-access coaxial CATV cables and so on. The various media, transmission rates, error characteristics and so forth will dictate differing packet sizes. In order to deal with evolving technology, we take the view that it should be possible for a subscriber to emit a datagram which, on passing through a gateway, can be "fragmented" into smaller pieces which can later be reassembled.

No matter what fragmentation strategy is employed, the destination subscriber must still be prepared to reassemble fragments produced by the last gateway. There is consequently little point in putting a reassembly function in the internet gateway. To formulate an abstract model of this view, we must modify figures 1, 2, and 3 to reflect a gateway layer of protocol for internetting.

As shown in Figure 4, a gateway between two networks is composed of an internet gateway part which exchanges routing, congestion control, status, and accounting information between networks and a local gateway part which is aware of the characteristics of the network(s) to which it is connected. Internet datagrams leaving the internet gateway may be fragmented by the local gateway, and incoming fragments may be re-assembled.



Gateway Functions
Figure 4

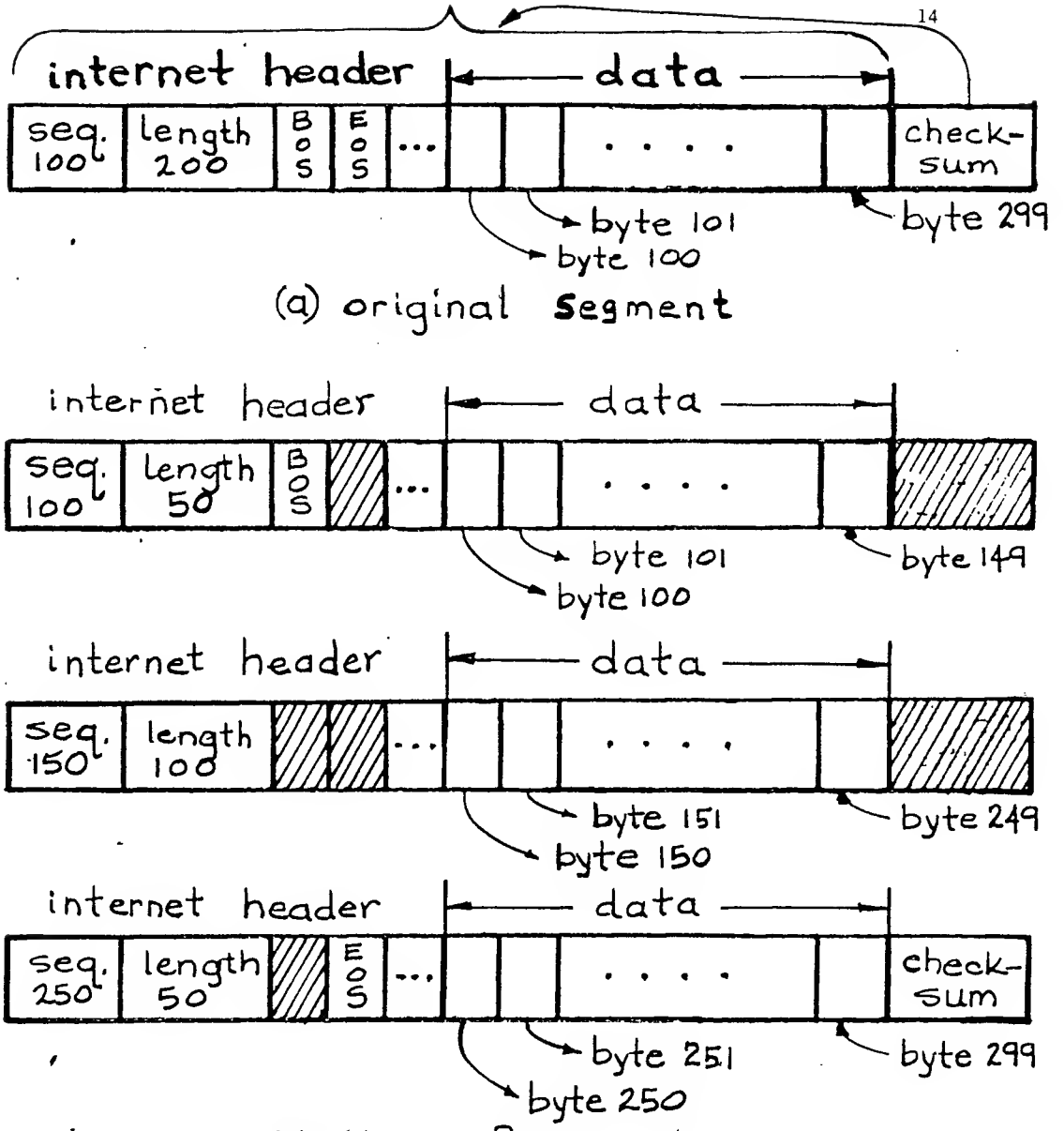
The destination subscriber's local gateway should be prepared to re-assemble datagrams which may have been multiply-fragmented or whose fragments have been routed through different internet gateways to get to the destination. This assumption allows for maximum freedom in fragmenting and routing datagrams through multiple networks. The datagram format must obviously accommodate this operation, while at the same time, allowing for intermediate re-assembly without loss of boundary marking information. We take it as essential, furthermore, that end-to-end checksumming over the datagram header and text must be possible so as to assure that datagrams delivered to the higher level protocol are without errors (to within a very small probability).

One conceptual strategy for achieving this goal is to adopt a datagram format which includes an end-to-end sequence number, length, checksum, and two flags meaning "beginning of segment" (BOS) and "end of segment" (EOS). As shown in Figure 5, it is possible to fragment and reassemble fragments while preserving an end-to-end check on the validity of the data as long as the BOS and EOS flags are properly handled.

In figure 5(a) we illustrate a datagram made up of an internet header and 200 bytes (8 bits/byte) of data. A checksum has been computed over the header and all data. Each byte of data is associated with a unique sequence number. The next datagram (not shown) will start with sequence number 300 (i.e., 100 + 200) to preserve this uniqueness. The two fragmentation control flags, BOS, and EOS, have been set, indicating that the datagram is complete and contains an end-to-end checksum for verifications.

If the datagram must be fragmented, as is shown in Figure 5(b), the internet headers are slightly modified to indicate which sequence numbers (i.e., which bytes) are contained in the data. The EOS flag is reset in all but the last fragment. The sequence number and length of each fragment's internet header have been modified to reflect the fragmentation. At an intermediate or destination local gateway, the fragments can be re-ordered using the sequence numbers and reassembled to verify the end-to-end checksum. The strategy works, even if the source local gateway retransmits the original datagram and it is fragmented differently as a result of passing through a different set of gateways because of the unique association of data bytes and sequence numbers. Fragmentation at any byte boundary is thus permissible.

The strategy outlined above allows for complete freedom to route and fragment packets passing through multiple nets, placing few constraints on subnet implementation. Minimizing these constraints will permit the interconnection of a wide variety of data networks and allows for adaptation to new transmission media and network organizations.



Datagram Fragmentation Schedule

Figure 5

Another aspect of the X.25 virtual circuit interface is of some concern for higher level protocol designers. Although it is not entirely certain, it appears that the X.25 interface would allow a network to accept several packets from a subscriber, form a larger subnet packet from these, and deliver them in one packet to the destination. At first glance this is attractive because it improves efficiency. However, if there is no mechanism for indicating the original boundaries on delivery, the receiving subscriber may be forced to search the incoming packet to find the relevant boundaries. Even worse, when certain kinds of privacy transformations are employed, the artificial coalescing of two subscriber packets on delivery can lead to deadlocks in which the transformations cannot be correctly undone. If subscribers using the X.25 virtual circuit interface are, in fact, going to be faced with this problem, it is clear that additional control information must pass from end-to-end within the network(s) to allow both subscribers to stay synchronized.

Conclusion 6: End-to-end encryption services may fail if the X.25 interface arbitrarily recombines packets before delivery to a host. Packet fragmentation and reassembly under X.25 appears to lose important end-end boundary information.

For example, if subscribers are trying to exchange packets containing subscriber level headers, but X.25 delivers these in arbitrary units, without any indication of the original subscriber boundaries, it may be necessary to employ in-band flag characters, and bit-stuffing (as in HDLC) to maintain data transparency. A properly formulated network interface would assure that useful subscriber boundary information is conveyed from end to end, eliminating the need for line-level protocol tricks at the host-host level of protocol.

New Technology and Services

As computers become more widespread in business applications and as their cost decreases, we can safely predict an increase in the demand for inter-computer communication services. Many services such as Electronic Funds Transfer, point-of-sale transaction management and credit card verification require only that short messages or packets be sent from one place to a variety of different places. Setting up virtual circuits to transmit a single message imposes needless inefficiency and overhead on the subscriber and makes the existing public and commercial packet switching services less attractive. The Bell Transaction Network uses a datagram interface, for example, precisely because it fits the application more appropriately.

If public or commercial packet networks are to support real-time services requiring low delay but not sequencing or guaranteed delivery, then the typical virtual circuit service must be modified. For example, a virtual circuit, because it is implemented using store-and-forward methods in a packet switched net, exhibits very good undetected bit error rate characteristics (less than 1 bit in a trillion). Each packet is error checked as it is forwarded from node to node and retransmitted if it is received in error. For added reliability, most virtual circuit services also employ end to end (DCE to DCE) retransmission to protect against packet loss within the network due to a node failure. Such reliability measures usually result in an increase in the variance of inter-arrival time of packets delivered to the destination subscriber.

As more computer terminal equipments are equipped with built-in microprocessor(s) and memory, the need for PAD services will be reduced because these terminals can perform their own packet assembly and disassembly. From the point of view of the network, the distinction between host and terminal will blur since both will be capable of sending or receiving packets to or from several remote sites.

Multidestination addressing, long in use in the business world (in the form of carbon or photocopies) and in the military world (particularly in Autodin I), has an obvious place in packet networks to support electronic mail, distributed file systems, distributed operating systems, voice conferencing (or mixed media conferencing) and so on. Some applications may even require that a particular packet or sequence of packets be delivered to all sites in the net. For example, some information sources (weather instruments, highway traffic sensors) may not know exactly which sites should receive data, so it might be broadcast to all of them.

If it were necessary to "set up" a virtual circuit to every possible destination before doing a multidestination broadcast, the potential delay and overhead to explicitly remember each destination at the source DCE would be very high. One alternative is to create "multi-destination" names. Intermediate nodes in the network would have routing table entries for these multiple-site names. Of course, these entries would have to be created and destroyed dynamically. Similarly, a particular name might be reserved to mean "all destinations".

Maintaining virtual circuit characteristics (reliable, sequenced data) in the presence of multi-destination broadcasting to many sites will be impossible, owing to the amount of acknowledgement traffic which would be required and the amount of table space needed at the source DCE to keep track of end-to-end acknowledgements.

Conclusion 7: Packet broadcasting is inefficient at best and impossible, at worst, if virtual circuits are used to achieve it.

Real-time Data Transfer Services (e.g., packet speech)

Although it is very unlikely that packet networks could support the volume of voice traffic that circuit-switched networks typically handle, a limited speech capability in packet nets may be useful to augment existing computer-based teleconferencing systems.

Experiments with "packet speech" conducted with support by the Defense Advanced Research Projects Agency, for example, have demonstrated the feasibility of carrying small amounts of compressed, digitized voice through the ARPANET. Indeed, advances in digital voice compression using linear predictive coding (LPC), continuous variable slope delta modulation (CVSD), and adaptive delta modulation (ADM), homomorphic and channel vocoders demonstrate that relatively low rate speech (on the order of 1000 bits/sec on the average with peak rates around 3000 bits/second) is feasible. If speaker silence is detected at the digitizer, then it is straightforward not to transmit during the silence periods. Packet networks do not require continuous transmission to keep both subscriber ends in synchrony and thus, the removal of silence becomes straightforward.

For compressed packet speech to work well, it is far more important to achieve low delay and low inter-arrival time variance than ultra-reliability, provided that a sufficient number of packets actually do make it through. Some adaptive delta modulation schemes have been tested at packet loss rates as high as 10% with results equivalent to a bit error rate of under 1%. In fact, if the source subscriber can provide suitable time-stamps on each compressed speech packet, it is desirable not to attempt to resequence packets at the destination DCE but to let the destination subscriber decide, based on the timestamps, whether to "play" or discard the incoming packet, or to buffer it in the hope that a missing packet will arrive soon enough that both can be "played". The conclusion, in this instance, is that compressed, packet speech, integrated with data services, requires a different mode of operation than typical virtual circuit systems offer.

Conclusion 8: Real-time applications requiring low delay but not requiring guaranteed or sequenced data delivery will be inefficient and possibly inoperable through X.25 virtual circuits.

Conclusions

We have shown that the packet network virtual circuit interface concept, as characterized in the CCITT X.25 recommendation does not satisfactorily meet all forseeable packet communication requirements of commercial, public, private, and military networks.

In addition to the eight conclusions developed thus far it is quite apparent that to achieve reliable and controllable end-to-end communication at the host level, it will be necessary to implement end-to-end flow control at that level. Furthermore, for logical messages larger than the maximum "packet" accepted by the network to which the host is attached, fragmentation and reassembly procedures will be needed. These will require some form of sequence numbering to assure reliable communication in the event of virtual circuit "resets" within the packet network, hosts must be prepared to retransmit packets which have not been acknowledged by the destination host (or for which the network has reported "non-delivery"). This latter requirement implies that the host must also provide duplicate detection facilities. All these facilities would also be needed if the host were to use a datagram service rather than a virtual circuit service. We are therefore led to a final conclusion:

Conclusion 9: Regardless of the use of virtual circuits or datagrams, host level protocols must provide sequencing, retransmission, duplicate detection, fragmentation, reassembly, and flow control techniques to assure reliable and controlled host-level intranet and internet communication.

For reasons of national and international security, it is imperative that the standards set for public and international packet networking accommodate both public and military requirements.

Recommendations

The addition of a datagram interface mode has been proposed in the past and several alternatives have been offered for its realization, maintaining compatibility with the current X.25 recommendation.

It is essential that a datagram mode of operation be included in the X.25 packet network interface standard to correct the deficiencies outlined above and to generally accommodate the evolution of computer communications technology.